グラフィクスとＣＡＤ　58－5
（１９９２．８．１０）

パソコンによる三次元自己相似形図形処理に関する一試み

森　久紘

滋賀女子短期大学

リカーシブと相対座標（タートル・グラフィックス等）による三次元グラフィックスを試みた。この方法はすでによく使われている手法であるが、三次元のものは多くない。そしてこれによるプログラムはほとんど数式が不用で比較的簡単にプログラムできる。
繰り返し相似形図形を描がかせることにより、無限の空間と時間を感じないでもない。

A Study on Similar Figures of 3－D Computer Graphics by Personal Computer

Hisahiro Mori

Shiga Woman's Junior College

This paper tries to show an example of 3－D graphic in recursive and turtle graphics method. This method is rather popular recently but not so many examples in 3－D method can be seen even now. Programming in this method is relatively easy, because it does not need equation. By repeated drawings of similar figures. It is not impossible to feel infinite time and space.

＊はじめに

青海波という模様があるが、これは三重に描いた円弧を魚の鱗状に連続した柄であり、重なる波を表現したものである。九鬼周造は『青海波』と題する随筆のなかで、１）同名の音曲の話ではあるが『見渡す限り波また波。無限の重畳そのものがとてもすばらしい。・・・・無限な時間は青い色の波、波、波、波、波、波の重畳また重畳にほかならぬ。・・・』といっている。この場合は相似形が規則的に連続した美しさである。

森政弘は随筆『心眼』のなかで『不思議な世界を、ありありと自分の目で見る方法がある』といって、合せ鏡に映つた自分自身がまた鏡の中に無限に映っている状態を説明して、『文字どおり無限にきみがいる』といっている。２）無限に同じ姿が鏡の中へ中へと連綿と重なっている不思議である。

日本には造形的に完成していると思われる家紋というものがあるが、３）　そのなかに『子持ち』という構成のものがあり、図１のような『子持ち分銅』は親の『分銅』と相似形の子の『分銅』が左右にはめ込まれている。まさに自己相似形図形である。このような図形を描くのに再帰的呼び出しおよび相対座標系（タートル・グラフィックス等）を利用すると、数学的な知識がなくとも比較的簡単に自己相似形図形が描けると思われるので紹介したい。なおここでいう自己相似形図形とは所謂フラクタル図形の定義に必ずしもあてはまらない場合もあるのでお断りしておく。

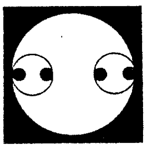

図　　１

＊タートル・グラフィックスについて

この機能はＬＯＧＯ言語などに採用されいいるもので、タートル（亀）という一種のカーソルをロボットのように移動させて移動軌跡を線分によって画面上に表示する方法である。例えば一辺１００の正方形を描きたい場合、通常、座標値（０，０）から描きはじめるとすると、（０，０）、（０，１００）、（１００，１００）、（１００，０）、（０，０）と順に結んでゆく。これは絶対座標値であるので同じ図形は一つしかない。では次の方法としてタートルの向きは変えないで、前へ１００、右へ１００、後へ１００、左へ１００と移動させても描くことが出来る。この場合は相対座標値である。それではタートル・グラフィックスではどうするかというと、タートルを前へ１００進めて右に９０度方向を変え、また前へ１００進めて右へ９０度方向を変えるという具合にこの操作を４回繰り返せばできあがる。

＊基本的な関数（サブルーチン）

基本的な三次元グラフィックス機能をもった関数（サブルーチン）群があれば有用であるので作成した。これには先記の三通りの描画の機能を含んでいる。タートル・クラフィックスは通常、二次元であるが三次元に拡張したもので、基本的なアルゴリズムはｉｎ－ｆｏｒｍａｔｉｏｎ誌１９８４年６月号の原田康徳、鏡慎両氏による『トンボグラフィックス』によっている。４）では簡単に一部の関数の機能の説明をする。

homein( )：タートルの方向を北向きにして、座標値（０，０，０）に移動設定する。

eye(od,er,eh)：視点の座標値を設定する。

henza( )：絶対座標値をＹ軸、Ｘ軸、Ｚ軸を中心軸としてｙｒ度、ｘｒ度、ｚｒ度の順に回転する。

colr(c0,c1,c2)：Ｒ，Ｇ，Ｂの値を指定する。

cor(col)：引数の値を－１にするとタートルの移動時に軌跡を残さない。正の値のときは軌跡を残す。

tn(t)：タートルを角度ｔ度右回転させる。ｔが負の時左回転。

sp(s)：タートルをｓ度、ネジの締まる方向へ回転させる。ｓが負の時ゆるむ方向。

up(u)：タートルの頭部を手前へ起こすような方向へ回転させる。角度はｕ度である。

mvto(xpp,ypp,zpp):絶対座標値（ｘｐｐ，ｙｐｐ，ｚｐｐ）へタートルを移動させる。
mvto2d(xpon,ypon)　：二次元の（ｘｐｏｎ，ｙｐｏｎ）の座標値にタートルを移動する。
mv(m):距離ｍだけタートルを前へ進める。
mr(m):タートルを右へｍだけ移動させる。引数の値が負のとき左方向へ移動する。
mu(m):タートルを手前（頭上）方向へ距離ｍ移動させる。ｍが負のときは足元方向へ移動。
rfu(r,f,u):　タートルを右へｒ、前へｆ、手前へｕの位置へ移動させる。
gponf(pg):タートルの移動軌跡を配列に数値として残す場合は引数を１または２、残さない場合は０とし実行する。１の場合は相対座標値、２の場合は絶対座標値となる。
grapd(gg):配列に蓄えられている座標値に従って順次タートルを移動（再現）させる。
prow( ):配列のデータをディスクに書き込む。
progr( ):　ディスクのデータを配列に読みこむ。
wbai(x,y,z):　絶対座標値の倍率を設定する。
rbai(r,f,u):相対座標値の倍率を設定する。
tbai(m):タートル・グラフィックス時の移動距離の倍率。
以上のような関数群を作成した。５）６）
なお、先にも述べたがこれら関数群のタートル・グラフィックスの機能はＬＯＧＯ言語がもっているものである。特に三次元のＬＯＧＯである『３Ｄ－ＬＯＧＯ』（ユニー・バイナス）やＭａｃ用のＥｘｐｅｒＬＯＧＯがすでに市販されていることをお断りしておく。７）

＊簡単な曲線の図形作成

タートル・グラフィックスでは円弧を描く場合、方向変化の角度を一定、進ませる距離も一定で数回繰り返すとできる。では螺旋はどうであろうか。初期値に順次、加算または減算させた変化によって描く。あるいは乗除させる場合、平方根の値による変化などが考えられる。そして変化させる対象は、進ませる距離、方向変化の角度が考えられる。例えばアルキメデスの螺旋はタートルを進ませる距離に順次一定の値を加算すれば描ける。乗算すれば対数螺旋になる。進ませる距離は一定にし、こんどは方向変化の角度の値に順次一定の値を加算するとクロスソイド曲線となる。８）ｍａｉｎ（　）からｅｎ（ｉ）を呼び出して実行させるとするとプログラムは次のような簡単なものである。

```
en(i)
float i;
{
    if(i>=1600.0) return;
    mv(20.0);tn(i);
    en(i+1.5);
}
```

これを少し立体的（三次元）にするには線分を描くたびに少し頭を起こさせてみよう。

```
en(i)
float i;
{
    if(i>=1600.0) return;
    up(3.5);mv(20.0);up(-3.5);tn(i);
    en(i+1.5);
}
```

プログラムは上記のようになり、図２のような図形が描ける。また下のようなプログラムを実行すると図３や与える数値によっては波形となったりする。

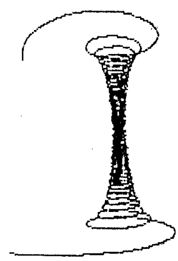

図　２

```
en(f);
float f;
```

```
{
  float ff;
  ff=cos(f*3.14159/180.0)*50.0;
  tn(ff);mv(10.0);
  en(f+18.0);
}
```

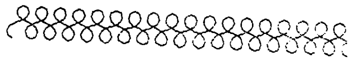

図　3

このようにタートル・グラフィックスではリカーシブと組み合せることにより、加減乗除程度の計算で短いプログラムによって図形を描ける可能性があるように思われる。8)

＊自己相似形図形の作成について

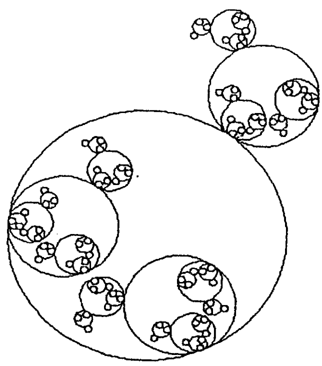

図　4

図4のように円のなかに子供の円が二個と外側に一個接している（子持ちの円）図形は下のようなプログラムで描くことが出来る。

```
enen(m)
float m;
{
     if(m<0.5) return;
     en(m);
     tn(180.0);enen(m/2.5);tn(-180.0);
     en(m);
     enen(m/2.5);
     en(m);
     enen(m/2.5);
}
en(m)
float m;
{
  int i;
  for(i=0;i<12;i++)  { tn(10.0);mv(m);}
}
```

ｅｎ（ｍ）は三分の一の円弧を描く部分である。６、８、１０行ではリカーシブ・コールしている。６行目ではタートルを１８０度右回転させてからリカーシブ・コールしているので円は外をむく。これとよく似た図形を立体的（三次元）に描がせてみよう。（図５）

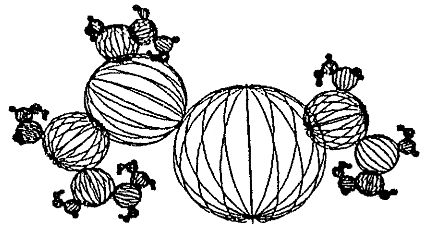

図　5

```
en3d(m)
float m;
{
   int j;
   if(m<0.5) return;
   tn(-5.0); en(m);
     tn(180.0);
       en3d(m/1.5);
     tn(-180.0);
   en(m);en(m);tn(5.0);
   for(j=0;j<8;j++)   {
     up(20.0);
     tn(-5.0); en(m);em(m);en(m);tn(5.0);
   }
   tn(-5.0); en(m);
     tn(180.0);
       en3d(m/2.5);
     tn(-180.0);
   en(m);en(m);tn(5.0);
   for(j=0;j<10;j++)  {
```

```
      up(20.0);
      tn(-5.0);en(m);en(m);en(m);tn(5.0);
    }
}
```

上のようなプログラムになり、ｅｎ（ｍ）は二次元の場合と同じである。リカーシブ・コールの行は子供の球を描く部分である。その他の行は描きはじめる方向の角度設定もしくは球の部分の円弧を描く部分かである。
図６、図７は同じ要領でリカーシブとタートル・グラフィックス等を用いて描いたものである。

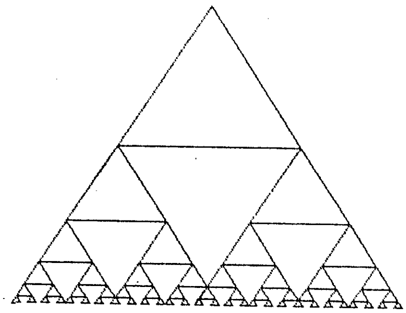

図　６

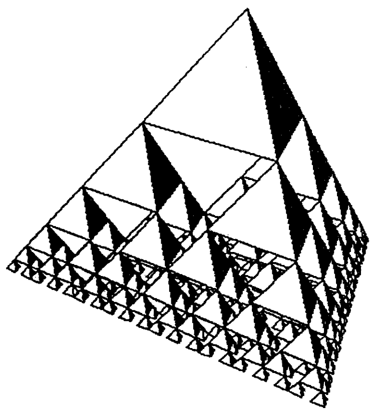

図　７

図６のプログラムは下記のようになる。

```
sank(m)
float m;
{
   if(m<2.0) return;
   cor(1);sankaku(m);cor(-1);
   mv(m);sank(m/1.84);tn(120.0);
   mv(m);
   tn(-120.0);sank(m/1.84);tn(120.0);
   tn(120.0);
   mv(m);tn(120.0);
}
sankaku(m)
float m;
{
   mv(m);tn(120.0);
   mv(m);tn(120.0);
   mv(m);tn(120.0);
}
```

また図７のプログラムは次のようになる。

```
sank(m,n)
float m;
int n;
{
   float bai;
   bai=1.62;
   if(n==0) return;
   cor(1);sui(m);cor(-1);
   rfu(m*0.5,m*-0.8,m*1.73205/6.0);
     sank(m/bai,n-1);
   rfu(-m,0.0,0.0);
     sank(m/bai,n-1);
   rfu(m*0.5,0.0,-m*1.73205/2.0);
     sank(m/bai,n-1);
   rfu(0.0,m*0.8,m*1.73205/3.0);
}
sui(m)
float m;
{
```

```
  suis(m);sp(120.0);
  suis(m);sp(120.0);
  suis(m);sp(120.0);
}
suis(m)
float m;
{
  rfu(m*0.5,-m*0.8,m*1.73205/6.0);
  rfu(-m,0.0,0.0);
  rfu(m*0.5,m*0.8,-m*1.73205/6.0);
}
```

となり、ｓｕｉｓ（ｍ）で三角形をつくり、ｓｕｉ（ｍ）で三角錐をつくっている。

以上のように二次元の場合と全く同様の感覚でプログラムを作成することができ、またプログラムと図形との対応関係が対になっており、直感的に把握しやすいものと思われる。

＊基本的構成構造について

上記プログラムをみていると基本単位となる図形を描かせる部分と、その基本単位を描く方向（角度）、また場合によってはその位置を示す部分によって出来あがっていると考えられる。

日本の家紋の構成は様式的に決まったものが多数あり、たとえば『並び』（図８）と称するものは二つの紋が左右に同じ向きで並んだものである。

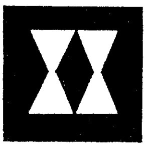

図　８

したがってタートルが北を向いているものとして、左、右の順に描くとすると、基本単位となる図形を描き、関数ｍｒ（ｍ）によって距離ｍだけ移動し、また基本単位となる図形を描くということになる。ただし基本単位の図形は北を向いた方向から描きはじめ、描き終ったときも北を向き、描きはじめと同じ位置に戻っているものとする。このように考えると『並び』の構成構造はｍｒ（ｍ）ということになる（描いたのち、タートルが元の位置に元の角度に戻っていなければならないとすると、もう少し複雑になる）。

もう一例あげてみよう。『寄せ』（図９）という構成があるが、まず中心よりｍタートルを進めて、基本単位の図形を描いてｍ後退させ、右に１２０度回転させるという動作を三回繰り返すということになる。９）

図　９

プログラムでは

```
for(i=0;i<3;i++) {
      mv(m);
      基本単位の図形を描く；
      mv(-m); tn(120.0);
  }
```

ということになる。したがって放射状に三方の線分があるような構成ということになる。

先ほどの図７の図形において、プログラムのｓｕｉ（ｍ）は三角錐（基本単位の図形）を描く部分であるから、その部分を取りのぞき表示させると、図１０のようになる。これがこの図形の構成構造の一つを表しているといえなくもない。そしてこのような構成構造が美の表現の一つの秩序ではないだろうか。

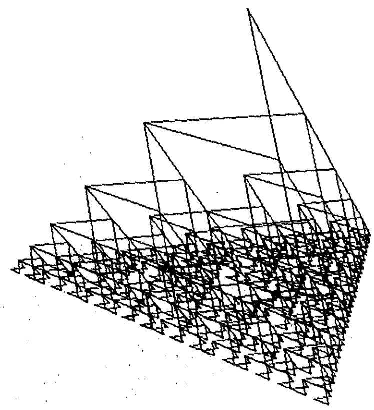

図　１０

よく再帰による図形の例として枝分かれした木が登場する。三方向に分かれた木は次のようなプログラムになる。

```
tree(m,n)
float m;int n;
{
    if(n==5) return;
    mv(m);tn(45.0);
    tree(m/1.5,n+1);
    tn(-45.0);
    tree(m/1.5,n+1);
    tn(-45.0);
    tree(m/1.5,n+1);
    tn(45.0);mv(-m);
}
```

これを三次元の木にするとプログラムは、

```
tree(m,n)
float m;
int n;
{
    if(n==5) return;
    mv(m);
    tn(45.0);
    tree(m/1.5,n+1);
    tn(-45.0);sp(120.0);tn(45.0);
    tree(m/1.5,n+1);
    tn(-45.0);
    sp(-120.0);sp(-120.0);
    tn(45.0);
    tree(m/1.5,n+1);
    tn(-45.0);sp(120.0);
    mv(-m);
}
```

となるが、これを基本的な構成構造とみなして、基本単位となる色々な図形について、これをもとに描かせてみると、たとえば図１１、図１２、図１３のごとくになる。　これらはいずれも構成構造は同じであり、この構造自体が美的秩序の一つの要素と考えてはどうであろうか。

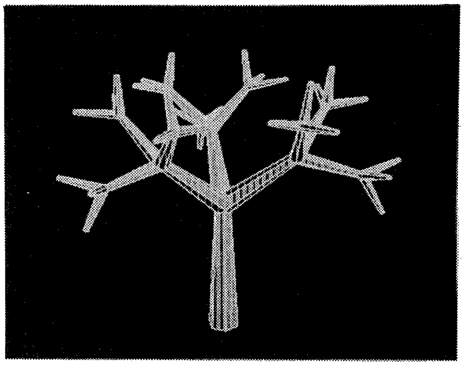

図　　１１

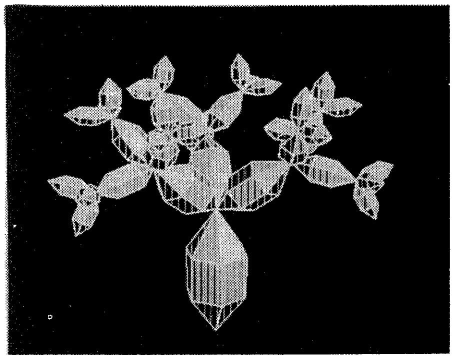

図　　１２

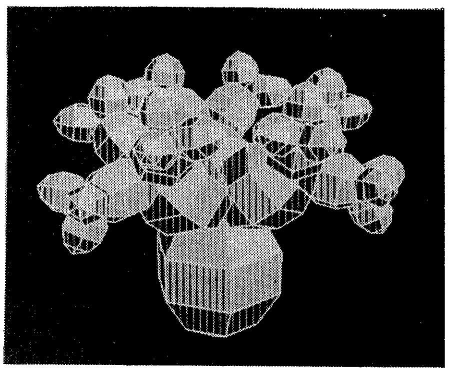

図　　１３

つづいての例で図１４、図１５は立方体を基本単位の図形とし、二方向へ角度を変えて描かせたものである。

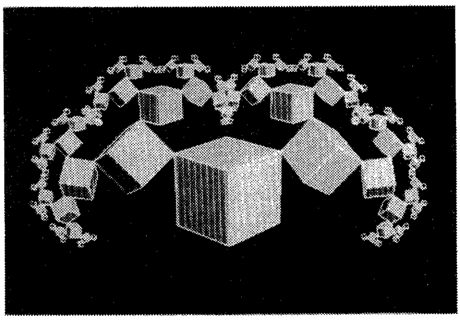

図　１４

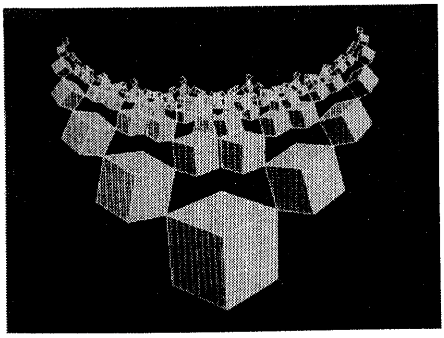

図　１５

図１６は四方向へ立方体を描かせたもので、角度を内、外に変えている。

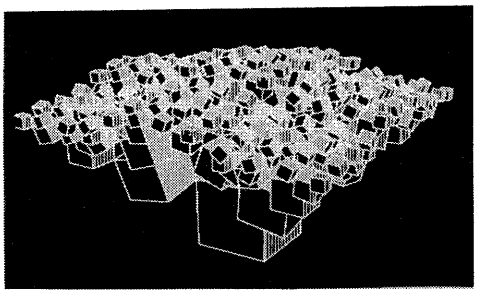

図　１６

＊おわりに

まず始めにタートル・グラフィックスというものを取りあげ、加減乗除等の簡単な式によってどのような曲線が描けるかを紹介した。次にこれとリカーシブ・コールによって二次元図形だけでなく三次元の立体図形の相似形図形による繰り返しによるやや複雑な図形が、短いプログラムで比較的わかりやすくプログラムできることを述べた。最後に基本的な単位となる図形が繰り返し表示される図形において美的な構成基準となりうる骨組があるのではないかと考えた。

参考文献

１）．九鬼周造：をりにふれて、岩波書店、１９４１．
２）．森政弘：心眼、佼成出版社、１９７６．
３）．吉野竹次郎：紋の泉、洛東書院、１９２６．
４）．原田康徳・鏡慎：３次元版タートルグラフィックス・トンボグラフィックス、ｉ－ｎｆｏｒｍａｔｉｏｎ・インフォーメイションサイエンス、６月号、Ｐ５９－Ｐ６５、１９８４．
５）．森久紘：ＬＯＧＯによるパソコン・グラフィックス・ワイヤーフレームによる三次元表現、滋賀女子短期大学研究紀要、１９８７．
６）．Ｂ．Ｗ．カーニハン・ＤＭ．リッチー：石田晴久訳：プログラム言語Ｃ・ＵＮＩＸ流プログラム書法と作法、共立出版、１９８１．
７）．川野洋：ＬＯＧＯ入門、電子計算機プログラミング＝１０、培風館、１９８７．
８）．森久紘：再帰的呼び出しとタートルグラフィックスによる図形処理プログラムについて、滋賀女子短期大学研究紀要、１９８９．
９）．森久紘：相対座標による図形構成構造のプログラミングについて、滋賀女子短期大学研究紀要、１９９０．